Learning Subjective Relevance to Facilitate Information Access

James R. Chen* & Nathalie Mathé†
NASA Ames Research Center
Moffet Field, CA 94035-1000
jchen@ptolemy.arc.nasa.gov, mathe@ptolemy.arc.nasa.gov

Abstract

As the amount of available electronic information is dramatically increasing, the ability for rapid and effective access to information has become critical. Most traditional information access methods rely on measures of relevance based on information content. We propose a new approach which augments existing information access methods with subjective relevance learned from user feedback. We developed an adaptive system which helps users access information by employing learned knowledge about which documents are likely to be relevant, given the current user's information need and user profile. This system is based on a model, called a *relevance network*, which learns and generalizes relevance information in a rapid, cost-effective, and incremental manner. We present the design of the relevance network and results of experimental evaluation.

1 Introduction

We are currently witnessing an explosion in the amount of information that is available on-line. This has created the need for new tools to assist people in quickly and effectively locating information. There are several ways to access information: traditional information retrieval [13, 7] builds a content-based index of all the documents; model-based information retrieval [3] requires a knowledge representation structure of the information domain; browsing (e.g., Mosaic) lets the user follow pre-authored information structures or hyperlinks. While these approaches add tremendous power to information access, they require extensive a priori construction of domain specific index information, hence are not always suitable for quick access to various information sources with dynamically changing individual needs.

We propose *relevance network*, an adaptive information access model independent of specific information domains. The model does not rely on customized data organization or content-based indexing. It learns subjective relevance information from user feedback for individual users or particular user groups. Information preferences of specific queries are memorized rapidly, and generalized over time for future retrieval with similar queries. As a component within a large information access system, the relevance network provides user-oriented customization facilities, which modify and filter relevance information provided by other means of retrieval.

2 Related Work

Relevance feedback methods in conventional information retrieval [13, 9] improve immediate retrieval performance by modifying the current query, based on user feedback on previous retrievals and existing domain information. Similarly, other adaptive information access methods often rely on specialized retrieval structures (semantic index) whose parameters get tuned by use, and require a large amount of a priori knowledge [8, 4]. Connectionist approaches require an extended training period to attain a state of fertility [1, 5]. The relevance network, in contrast, establishes relevance information directly through user feedback, without resorting to pre-defined knowledge or extensive training.

Adaptive interactive systems [14] is a very active research area. The canonical knowledge-intensive user modeling approach requires a priori encoding of both user and domain models [15]. These models are costly and difficult to acquire, and cannot easily be updated or customized during use [2]. Other approaches rely on individual user models dynamically learned by observing user's behavior [6]. These methods need large data collections before the models become useful.

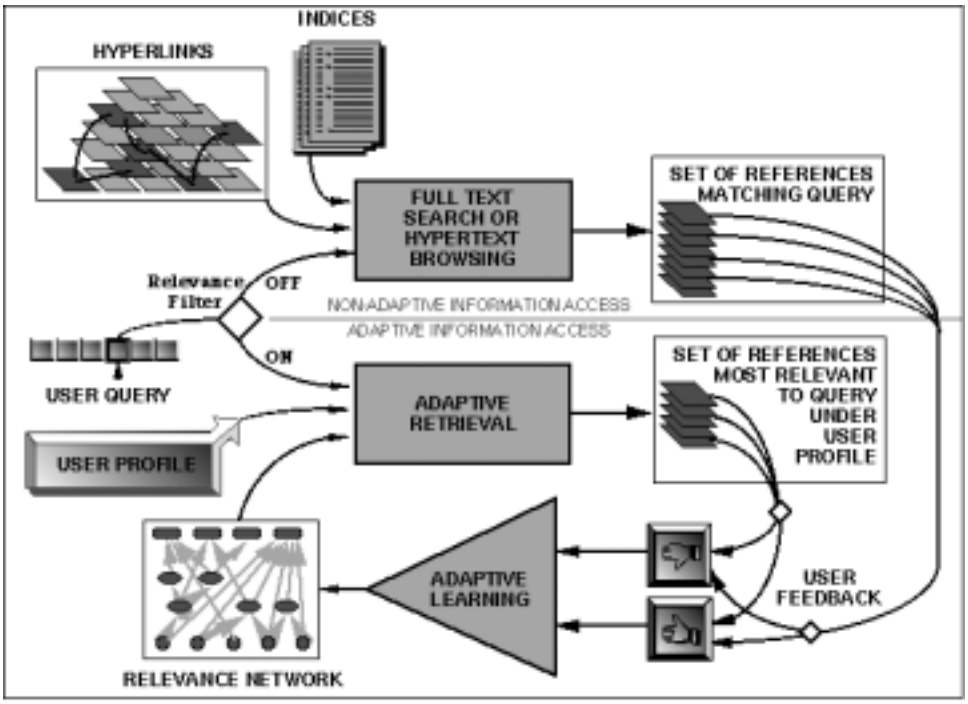
Our relevance network illustrates the idea of a learning agent applied to information filtering [11]. It is more closely related to the adaptive hypertext navigation approach proposed by Kaplan et al. in [10]. Their model memorizes user preferences in an associative matrix, but does not generalize the information for different usage.

3 Adaptive Information Access

We give an overview of our method for adapting information access to individual users profiles. Our method utilizes a relevance network which does not rely on any a priori knowledge, and interactively learns information relevance from end-users during information access. In a single user mode, the system provides faster access to hundreds of documents. In a multi-user collaborative mode, the system supports the creation of a corporate memory about documents

*contracted through Recom Technologies
†contracted through San Jose State University Foundation

Figure 1: *Adaptive information access overview*



relevant to particular user groups (a group's network), and it reduces the learning time for novices. It fosters collaboration among users by letting them access and share documents found relevant by other users with similar profiles. To bootstrap the network learning curve, the network can be initialized with a sample set of documents returned by a preliminary call to a traditional search engine, or better yet using a colleague's network, or a group's network.

The model is intended to work in conjunction with other information retrieval or hypertext systems, which provide non-adaptive information access (Figure 1). Users access information by executing an explicit query (resp. by selecting a hypertext link). Non-adaptive information access is performed by a search engine using stored indices (resp. an hypertext system using stored hypertext links). Adaptive information access is performed by our adaptive system using knowledge stored in the relevance network, pertaining both to indices and links.

The following scenario describes a typical information retrieval interaction with our adaptive system [12]. The user first specifies a query and relevance filter. The query is composed of a list of keywords (e.g., "telescope", "solar array") selected from the document full text index. The relevance filter is composed of a user profile, or list of tasks used to personalize the search for relevant documents (e.g., "astronaut", "on-orbit", "repair"). The user can choose to use his/her personal relevance network, one of the networks published by his/her colleagues, or one of the group networks which integrate feedback from several users. The last two options support knowledge sharing among users, by letting them access adaptations done by others.

Using learned knowledge stored in the selected relevance network, the adaptive system retrieves and displays a ranked list of references likely to be relevant, given the current user's query and profile. These references were either found relevant earlier for the same query and user profile (memorization function of the network), or are derived from references found relevant for similar queries or user profiles (generalization function).

When the user finds an interesting reference, she/he marks it as relevant by giving positive feedback to the system. The tasks for which this reference is considered relevant may be similar to the tasks used to access the reference, or can be specified by the user upon feedback. Users may provide feedback for any reference, retrieved with the network, or accessed through other non-adaptive methods of retrieval. This prevents the system from narrowing its suggested list of references over time. The adaptive system then memorizes in the user's relevance network that this reference was found relevant under the given query and feedback profile, and generalizes this relevance information for future queries.

In the following section, we describe the structure of the relevance network, its learning and retrieval methods, and the management of a dynamic network architecture.

4 A Compositional Relevance Network

A compositional relevance network models user preferences on information relevance with respect to given tasks. This network provides a domain independent information architecture which facilitates incremental storage of both relevance information provided by users, and relevance information computed through other traditional retrieval techniques. The network memorizes information on the relevance of references based on user feedback for specific queries and profiles. It also aggregates and generalizes such information to facilitate future retrievals with similar queries and profiles.

4.1 Relevance Network

A relevance network records measures of relevance of output nodes with respect to input nodes. For information retrieval purposes, an output node corresponds to a reference, which can be a document or any marked location within a document. There are two types of input nodes: basic and composite nodes. A basic input node corresponds to a descriptor. A descriptor can be a keyword in the index, a sequence of words in the text, or a user-defined task in a user profile. A descriptor can also be a reference, in order to retrieve other related references. A composite input node corresponds to a combination of query descriptors. Composites nodes are defined in section 4.2.

Figure 2: *An example of a simple relevance network*

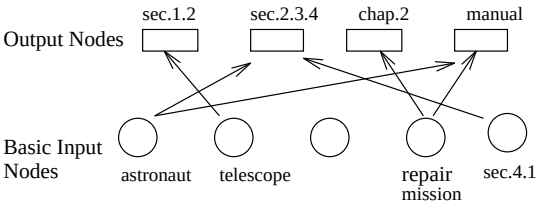


Figure 2 shows an example of a simple relevance network with only basic input nodes. Nodes in the top layer represent output nodes. Nodes in the lower layer represent input nodes. A user query is interpreted as an input activation pattern by the relevance network. A Boolean activation value of an input node denotes whether the corresponding descriptor is a member of the current query. An activation value on an output node denotes the relevance of the corresponding reference, conditioned by the current user query encoded in the input layer.

Associated with each connection from an input node to an output node is a relevance measure between the corresponding descriptor and reference. A network is initially empty[1]. As a user specifies queries and provides positive or negative feedback on the relevance of retrieved references, input and output nodes that do not exist yet in the network are created, and relevance measures associated with the connections are adjusted accordingly. A relevance measure, in its simplest form, is defined as the relative frequency of positive user feedback for a reference given a descriptor. Each relevance measure is maintained as two parts of a fraction: the number of positive feedback, S, over the number of total feedback, N. That is, a relevance measure R_{ij} of a reference j with respect to a descriptor i is

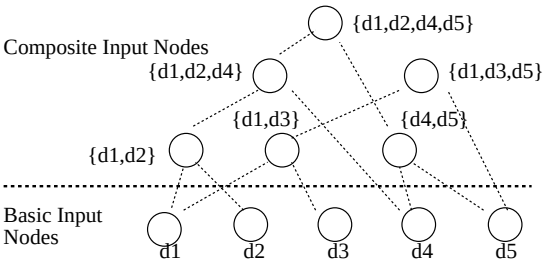
$$R_{ij} = \frac{S_{ij}}{N_{ij}} = \frac{NumberOf(PositiveFeedback)}{NumberOf(Feedback)}$$

Maintaining the total number of feedback in the denominator facilitates an accurate recording of both the relevance of the reference and the sampling precision of such relevance.

4.2 Composite Nodes

[1] When the relevance network is empty, references relevant to a query can be accessed through other retrieval means provided by the application interface. A relevance network can also be initialized using information attained with other retrieval techniques.

Figure 3: *Input layer of a relevance network with composite nodes. Links denote subset relations. Output nodes and relevance connections are not displayed.*



The relevance model described thus far records only relevance information based on single descriptors. User preferences for particular composite queries cannot be saved, and non-trivial relations[2] between references and descriptors cannot be encoded. To retain better information from user feedback, the relevance network accommodates composite nodes in the input layer, as illustrated in Figure 3. It is assumed that the relevance information associated with a composite node is more specific than the information associated with its nested subset nodes or its basic input nodes (corresponding to its descriptors). Therefore, during retrieval, a user query is first matched against highest level composite nodes, rather than lower level nodes. The use of composite nodes enables the system to derive more accurate relevance measures learned from previous queries.

Composite nodes are added to the network in two ways. A new composite node, corresponding to a user query, is added to the relevance network when a user provides feedback upon retrieval. A second composite node addition method based on co-occurrence statistics of query descriptors is discussed in 4.5.3.

4.3 Learning Relevance Measures From User Feedback

When a user provides positive or negative feedback for a reference given the current query, this relevance information is memorized and generalized. To memorize feedback information on specific queries, relevance of the connection between the reference and the composite node corresponding to the query is updated. If such a connection does not already exist, a new connection is created and the relevance measure initialized[3]. If a composite node corresponding to the query does not exist, a new composite is created with associated relevance information derived from that of its components. The derivation algorithm is described in 4.4.2.

To derive generalized relevance measures for new queries in the future, nodes which are more general than the user query inherit feedback: relevance measures from all proper query subsets including basic input nodes are updated. We describe below how relevance measures are updated or initialized.

[2] e.g., a reference relates to $\{Apple, Computer\}$ but does not relate to $\{Apple\}$.

[3] The user can choose to give feedback on any reference s/he has access to (not only on these references retrieved from the network), thus automatically indexing this reference with the current set of query descriptors.

4.3.1 Updating Relevance Measures

As mentioned above, relevance measures from an input node can be adjusted either through direct user feedback from a query of the same composition as that node, or through feedback inherited from its superset composite nodes. Direct feedback provides more accurate information pertaining to the node than inherited feedback. To compromise between memorization and generalization, a weight constant integer $C \geq 1$ is added to the relevance feedback adjustment: if $C = 1$, inherited feedback is as important as direct feedback; and the relative importance of inherited feedback decreases when C increases. Relevance measures are updated as follows:

$$R_{new} = \frac{S_{new}}{N_{new}} = \frac{S_{old} + \lambda * \delta}{N_{old} + \lambda}, \text{ where}$$

$$\delta = \begin{cases} 1 & \text{for positive feedback} \\ 0 & \text{for negative feedback} \end{cases}$$

$$\lambda = \begin{cases} 1 & \text{for inherited feedback} \\ C & \text{for direct feedback} \end{cases}$$

As mentioned in section 4.1, maintaining relevance measure as a fraction provides additional information on the precision of the measure. To accommodate more recent changes into the relevance network, a maximal threshold on the denominator is specified. When the number of total feedback exceeds this threshold, the denominator is no longer incremented, instead, a momentum term is used in the calculation:

$$R_{new} = \frac{S_{new}}{N_{new}} = \frac{S_{old}}{N_{max}} * \alpha + \delta * (1 - \alpha),$$

where α is the momentum, $0 \leq \alpha \leq 1$, and N_{max} the maximal threshold for number of total feedback. Since N_{new} must equal N_{max} in this case,

$$S_{new} = S_{old} * \alpha + \delta * (1 - \alpha) * N_{max}$$

For consistency with the relevance adjustment formula where the denominator is smaller than the maximum threshold, the momentum is typically set accordingly as: $\alpha = (N_{max} - \lambda)/N_{max}$

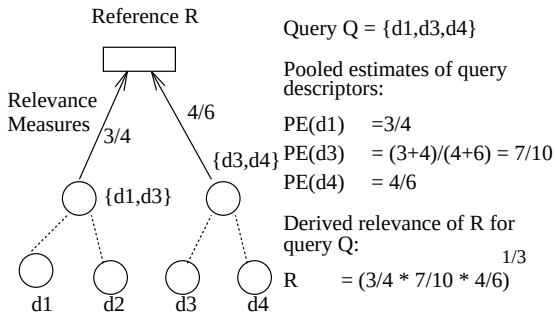
4.3.2 Initializing Relevance Measures

Relevance measure for a new connection is initialized with $S_{old} = 0, N_{old} = K$. K, a positive integer, corresponds to an initial negative bias. With this initial bias, the relevance measure asymptotically approaches one with the increase of the rate of positive feedback. The scale of relevance thereby provides better resolution for positive relevance information, and is biased against relevance measures with lower feedback frequency, which are assumed to indicate lower confidence of relevance accuracy.

4.4 Retrieval of Relevant References

In response to a query, the network retrieves and displays a list of references with the highest relevance measures. A query is assumed to be semantically constructed as a conjunction of member descriptors. Relevance measures are derived from nodes most specifically related to the query. Equal importance is given to all query descriptors. And for each descriptor, relevance measures from composite nodes are pooled together according to their statistical accuracy. This retrieval algorithm is illustrated on Figure 4, and detailed in the following paragraphs.

Figure 4: *An example of retrieval by derivation from subset query nodes for one reference*



4.4.1 Retrieval by exact match

Upon presentation of a query, if a composite node corresponding to the query already exists in the network, the relevance measures from that node to associated references are directly used to generate a ranked list of relevant references.

4.4.2 Retrieval by derivation

If a matching composite node does not exist, relevance measures from other input nodes, corresponding to proper subsets of the query, are used to derive a list of relevant references. For a relevance network with only basic input nodes, a simplistic estimate of relevance of a reference with respect to a query can be taken as the product of the relevance measures of that reference with respect to the descriptors of the query. The use of multiplicative estimation assumes no weighting information among individual query descriptors, and gives higher relevance to references of uniform relevance to all query descriptors.

With the presence of composite nodes corresponding to query subsets, relevance information is derived only from the top-level subsets, i.e., the ones not nested within other subsets. However, relevance measures for a query cannot be appropriately obtained by simply taking the product of top-level subsets relevance measures. Top-level subsets of a query can be of different sizes, and are not necessarily disjoint. References connected from one composite node may not be connected from another. Multiplicative measures, therefore, can be biased toward certain query components. We propose a heuristic derivation algorithm intended to provide impartial relevance estimations. For simplicity, we describe the relevance derivation algorithm for a single reference R, hence subscript reference indices are neglected in the formulae that follow.

For each individual descriptor in a query, a pooled estimate of relevance measure is obtained from the top-level subsets which contain that descriptor. Let $Comp_Q$ denote the set of descriptors in the user query Q, and S_{top} the set of top-level subsets of Q. The pooled estimate of relevance for a descriptor d_i in $Comp_Q$ is

$$PE_i = \frac{\sum_j S_j}{\sum_j N_j},$$

where the sums are taken over all j where $Comp_j \in S_{top}$

and $d_i \in Comp_Q$

Relevance measure of reference R for the query Q is then computed as

$$Relevance_{RQ} = \sqrt[n]{\prod_{i \in Comp_Q} PE_i},$$

where n is the size of $Comp_Q$.

The derived relevance measures are then used to generate an ordered list of suggested references.

In a query derivation where no subset composite node is present in the network, the algorithm degenerates to simple multiplicative derivation over basic input nodes. When applied to a query with disjoint top- level subsets, the algorithm reduces to taking the root of the product of all top-level subsets, each to the power of its own size.

4.4.3 Partial Match Derivation

To alleviate limitations imposed by the assumption of conjunctive queries, the network also supports the derivation of partial match results. Choices of the level of match are given to the users.

In partial match mode, the network generates additional references related to some, but not all descriptors of the query. In order to display these references in appropriate ranking relative to the fully matched references, a missing relevance value is introduced into the multiplicative derivation formula, as an estimate of relevance associated with a missing connection from a query descriptor to a reference. The value of missing relevance is also used in connection-trimming discussed in 4.5.4.

4.5 Managing Network Size And Capacity

4.5.1 Computing Cost and the Importance of Capacity Management

The primary cost of computing time in the use of a compositional relevance network is associated with the relevance derivation algorithm, which requires a search of composite nodes corresponding to all proper query subsets. In theory, this search can be computationally exponential with respect to the size of the query, due to the combinatorial large number of possible top level subsets. For pragmatic information retrieval purposes, however, the number of descriptors in a query is usually small, and only a very small percentage of all possible combinations of query descriptors are likely to be present in the network as subset composite nodes. Also, this cost of computing time is in the worst case linearly bound by the total number of composite nodes in the network. Thus the computational complexity of the derivation algorithm is not of realistic concern, provided that the number of composite nodes and the distribution of these nodes are well managed.

While the memorization capacity of a network increases with the number of composite nodes it contains, unnecessary composite nodes can potentially inhibit generalization of retrieval. Higher level large composites carry relevance information more specific to particular queries, whereas lower level smaller composites carry more general feedback information propagated from many queries. Relevance measures associated with larger composites, however, may also have less statistical accuracy since these composites receive less feedback from users. Managing the network capacity is therefore not only important in assuring control of the computing cost, but also important in maintaining a balance between the capacity of memorization and that of generalization.

4.5.2 Cutting Composite Nodes

A node cutting procedure is employed to control the size of a compositional relevance network. A maximal number of composite nodes allowed in a network can be specified. When a new composite node needs to be inserted, and if the network has reached its specified size limit, an existing composite node with the *least frequency of usage* is removed to make room for the new one. The frequency of usage of a composite node is calculated by recording the number of times the node is used in query execution. In addition, it is also incremented when the composite node receives direct user feedback. The purpose of this feedback-based usage update is to give more weight to composite nodes which carry information that cannot be easily derived.

The frequency of usage is a direct measure of how often a node is used for user queries. More subtly, it also serves as a measure of the *confidence level* of information accuracy associated with a node. The choice of removing nodes with the least frequency of usage ensures that the composites that remain in the network are the ones with most dependable relevance information.

4.5.3 Adding Composite Nodes

The query-based composite node creation method described in section 4.2 is intended to ensure quick learning of user preference by memorizing relevance information. Ideally, these nodes would also become useful components of the relevance network information structure, to derive relevance information for new queries. Unfortunately, smaller composites are less likely to enter the network since the relevance information they are associated with may be too general to be used as specific queries by users . Yet these smaller compositions may be of great importance for a network to assure effective encoding of relevance information.

A node creation method based on co-occurrence of query descriptors is devised to extract compositions important to the relevance network information structure. The network maintains a record of recently submitted user queries, and periodically generates statistics on sets of descriptors that often appear together in different queries. A simple formula is currently used to calculate co-occurrence statistics of query descriptors over a collection of queries:

$$C_s = \frac{f_s}{\sqrt[size(s)]{\prod_{i \in s} f_i}}$$

where C_s denotes the co-occurrence measure of descriptors in set s, f_s denotes the frequency of set s appearing in queries, and f_i the frequency of descriptor i in queries.

Composite nodes consisting of query descriptors of high co-occurrence statistics are then automatically added to the relevance network. Composite creation based on co-occurrence across queries facilitates effective encoding of non-trivial relevance information. It also helps generalize relevance information for future retrieval with similar queries.

4.5.4 Trimming Connections and the Scale of Relevance

Another measure of the network size is the number of relevance connections from input nodes to output references. Relevance connections are indexed in a database by the nodes they are associated with, and only the ones related

to a query need to be retrieved at a time. Unnecessary connections cause wasteful storage space, and can impact the performance of retrieval.

A connection with relevance value less than or equal to the missing relevance (described in 4.4.3, as a result of frequent negative feedback, is removed from the network. This connection trimming process prevents the network from unlimited addition of connections, and from keeping wasteful information of very low relevance. The relevance measures maintained by the network is therefore on a rational scale between the missing relevance and one, non-linearly proportional to the ratio of positive feedback. The use of a positive scale does not deprive the network of its capability of encoding negative relevance information. By trimming relevance connections from a composite more specific to a query, i.e., a query subset composite node of larger size, positive information carried by more general, smaller subset composites will be ignored. Thus the effect of negation is supported by the dynamic architecture of the compositional relevance network.

5 Experimental Results

The compositional relevance network is designed to model subjective indexing based on user preference of information access. It is intended as an information framework which integrates indexing structure provided by users, with indexing information generated by other conventional indexing methods and/or retrieval methods specific to the domain of application. For application purpose, it has been designed to be incorporated into large-scale, complex information systems. It is therefore difficult to test the full functionality of the network independent of the application domain. The user modeling perspective, in particular, requires extensive usage testing on a real-world system tuned for specific application purposes. As a first step, we focused on the validity of the proposed model and report on experimental results of the memorization capacity and generalization ability of the compositional relevance network, with no attempt to simulate user behavior. Real-world usage study, along with system integration[4], is in progress.

5.1 Simulation Setup

We used two test data sets of information retrieval from the SMART archive at the Computer Science Department of Cornell University. The first is a collection of 1963 Time Magazine news articles which consists of 425 articles and 80 queries. The second is the ISI collection of most highly cited articles and manuscripts in information science in the 1969-1977 period, with 981 articles and 76 queries. These experimental data sets were originally devised for the investigation of automatic indexing and document retrieval methods. Queries of the two sets employ large vocabularies, and extend a wide range of retrieval tasks. These are therefore not ideal for the testing of adaptive indexing, where higher similarities among queries more specific to individual users and/or task domains, as well as non-trivial relevance structures between references and queries are expected. The collections, nevertheless, were used here in our simulation experiments to ensure objective evaluation of the compositional relevance network.

Queries in these data sets are composed of common English phrases, e.g., "United Nation's efforts to get Portugal to free its African colonies". For our purpose, the queries are edited into sets of keywords with simplistic stemming, and common English words removed. Thus the above query becomes "unite, nation, portugal, africa, colony". The particular sequential order of words in a query is not utilized.

In simulation, a query is presented to the network as a set of descriptors, and the references retrieved by the network are compared with the target references listed in the original data set. Positive feedback is simulated for references that are in the target list but are not suggested by the network. Similarly, negative feedback is given to the network for suggested references not in the original data set. Although the compositional relevance network is designed to accommodate other means of retrieval, all simulation trials were conducted with initially empty network, to demonstrate clearly the functionality of the adaptive engine.

We first tested the memorization capacity of the compositional relevance network, i.e., the amount of relevance information a network can memorize with respect to the number of composite nodes. We then tested the generalization capability, i.e., the ability of a trained network to derive and suggest references for queries not previously presented to the network.

5.2 Memorization Capacity

The Time collection was first used to test the memorization capacity. Each query in this set should retrieve from one to 18 references. Query-based insertion of composite nodes was first disabled. Consequently the network did not contain any composite nodes hence could only encode and derive relevance information linearly[5] with respect to the basic descriptors. This network was trained with the complete set of queries in random order. For each query, positive feedback was given for all relevant references not retrieved, and negative feedback given for irrelevant retrieval of references not in the target set. After one complete cycle of training, i.e., each of the queries presented once, the relevance network was able to retrieve with a 100% recall[6], at a precision[7] of 93.1%. Specifically, all 321 relevant references, along with 37 irrelevant ones were retrieved. This result suggested that the data set is largely linear[8], hence the addition of composite nodes could make limited retrieval enhancement. Without enabling the query-based composite node insertion algorithm, five composite nodes of the highest co-occurrence statistics among the queries were added to the initial network. The modified network was able to improve the precision slightly to 94.8%, with 34 irrelevant retrievals, without affecting the 100% recall. The performance could not be improved further with more composites of lower co-occurrences added.

With the query-based composite insertion enabled, the network achieved perfect performance of 100% recall and precision, by memorizing the target references with 80 composites corresponding to the query set. When the composite node cutting procedure is in effect, the network was able to maintain perfect performance with as few as 10 composites.

Similar studies were done with the ISI data collection. In order to better demonstrate the network's capacity to

[4] The relevance network is currently being integrated into the Hyperman documentation system at NASA Johnson Space Center.

[5] A network without composite nodes is actually multi-linear due to the multiplicative relevance derivation algorithm, but can be converted to linear through logarithmic transformation.

[6] Recall is defined as the proportion of relevant materials retrieved.

[7] Precision is defined as the proportion of retrieved materials that are relevant.

[8] i.e., relevance information associated with a composite node can be derived from relevance associated with its members

encode non-trivial information, for this data set we eliminated query descriptors which appear in only one single query. Two of the 76 queries were invalidated consequently, as they became empty. The resulting set consists of queries of sizes ranging from 2 to 15 descriptors. Each query is to retrieve from 3 to 125 relevant references.

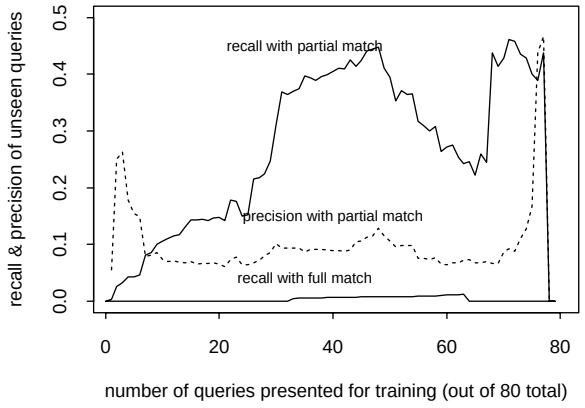
Different numbers of composite nodes based on levels of co-occurrence statistics were added initially. The results are shown in Table 1. Precision and recall statistics were collected after 1 and 3 training cycles. 100% recalls were attained for all simulation runs. The network with 41 composite nodes had higher total number of irrelevant references, yet better average precision than the network with 81 composites. This is because an average precision is taken over the precision measures of all queries, which is not the same as a pooled-average calculated directly from the total numbers of relevant and irrelevant references.

Table 1: *Comparison of retrieval results with different numbers of co-occurrence based composite nodes. 100% recall of 2655 relevant references were attained in all cases.*

number of composites	*average precision in %*		*total number of irrelevant references*	
	1 cycle	*3 cycles*	*1 cycle*	*3 cycles*
0	78.1	78.1	2176	2176
23	85.2	86.1	1142	964
41	88.7	88.9	716	694
81	87.7	88.9	664	512
179	90.8	92.6	402	298

5.3 Generalization

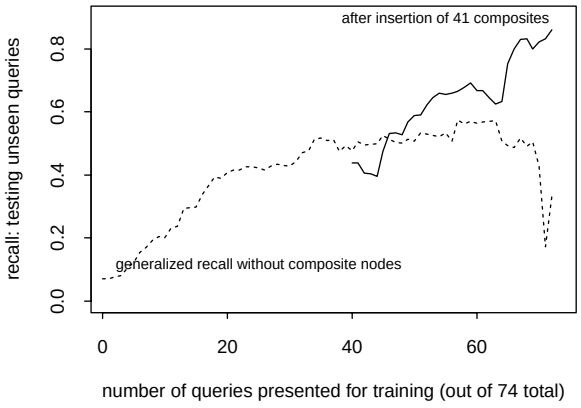
Figure 5: *Generalized retrieval results with unseen queries from the Time Magazine data set*



We first tested the generalized retrieval capability of the network on the Time collection data. Since this data set is largely linear, no composite nodes were employed. Simulation tests were conducted in both full match mode and partial match mode (described in 4.4.3). Queries from the set are presented to the network one at a time for feedback simulation. At the end of each query simulation, the remaining queries in the set, not yet presented to the network, were used to test the network's retrieval performance. The results are plotted in figure 5. With full match only, the average generalized recall was only near 1%. This was not surprising since many queries in this data set contained unique descriptors. The precision was not available since for many queries no reference was retrieved. With partial match, the recall increased to 40% with half of the queries presented. Recall statistics had wider variations at the end of the curve, as the sample size of test queries became smaller. The average precision in partial match retrieval remained mostly stable at around 10%.

We then tested the ISI data set for generalized retrieval in partial match mode. The curve of generalized recall was similar to that of the Time data set, with a peak recall at 57.5%. Precision stayed low at around 10%. To see the effect of composites on generalization, we ran another test with 41 composites of high co-occurrences inserted to the network after 40 of the 74 queries were trained, and the generalization test continued afterward. The generalized recall performance in this case showed consistent improvement as training continued, with a peak recall of 86% at the end. Figure 6 shows the curves of generalized recall with and without added composite nodes[9].

Figure 6: *Generalized recall results with unseen queries from the ISI data set, with and without added composite nodes*



5.4 Discussion

Simulation tests have shown that with query-based composite insertion, a compositional relevance network is capable of perfect memorization of relevance information based on user feedback. Test results also suggested that, while a network without composite nodes cannot maintain good precision of retrieval for data sets which contain non-trivial relevance information, the precision can be significantly improved with the addition of only a small collection of composite nodes. Composite nodes help improve precision through the provision of more specific information in the derivation of relevance. In a real-world application, the query-based composite insertion facilitates the customization of relevance information for specific queries of frequent usage, whereas the co-occurrence-based composite insertion helps the establishment of efficient information structure for long-term usage.

[9] The very low recall rate toward the end of the dashed-curve was partially caused by chance since only few unseen queries were left for testing.

These two composition methods, together with the node cutting procedure, work like a genetic algorithm which governs the evolution of the relevance network architecture.

We have shown also in simulation that the network is capable of generalizing retrieval of relevant references for queries not previously seen, through its feedback propagation algorithm. Generalized recall is further enhanced with the addition of composite nodes, which helps direct feedback information to appropriate composite structures, thereby releasing capacity of other parts of the network to encode more accurate relevance information.

Simulation with partial match also incurred low precision for generalized retrieval. This is partially due to the wide variation of relevance information of the test queries. Also, a complete list of references retrieved in partial match mode carries much additional relevance information, hence inhibits high precision. In practice, users are given flexible control of the amount of information displayed. Lastly, for simulation purpose, relevance information in a network was not initialized. The compositional relevance network, for application purpose, should be initialized with traditional or other domain specific indexing retrieval information.

To summarize, the compositional relevance network provides an information structure which facilitates integration of domain specific document indexing information and subjective user preferences. The adaptive network architecture and relevance connections support a balance between customization and generalization, while the control of balance between precision and recall is given to the users.

6 Conclusion

We have presented a model of subjective relevance for adaptive information access. The model employs a simple adaptive algorithm embedded in a dynamic indexing architecture based on user feedback. It does not require any a priori specialized index structure, nor any a priori statistical knowledge or computation. We have shown that with query-based composite insertion, a relevance network is capable of perfect memorization of relevance information based on user feedback. We have shown also that through its feedback propagation algorithm, a network is capable of generalizing retrieval of relevant references for queries not previously seen. While the query-based composite insertion facilitates the customization of relevance information for specific queries of frequent usage, the co-occurrence-based composite insertion helps the establishment of efficient information structure for long term usage. A relevance network can adapt to specific user needs, or it can generalize over multi-user information requirements, supporting sharing and collaborative work. The model can be easily integrated with traditional information retrieval methods to provide user-centered, rapid information access.

References

[1] R.K. Belew. Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. In *Proc. of the 12th SIGIR Conference*, pages 11–20, 1989.

[2] D. Benyon and D. Murray. Developing adaptive systems to fit individual needs. In *Proc. of the 3rd International Workshop on Intelligent User Interfaces*, pages 115–121, 1993.

[3] Baudin C., S. Kedar, J. Underwood, and V. Baya. Question-based acquisition of conceptual indices for multimedia design documentation. In *Proc. of the 11th National Conference on Artificial Intelligence*, pages 452–458, 1993.

[4] J.P. Callan, W.B. Croft, and S.M. Harding. The inquery retrieval system. In *Proc. of the 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[5] H. Chen. Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science*, 46(3):194–216, 1995.

[6] A. Cypher. Programming repetitive tasks by example. In *Proc. of the ACM Conference on Computer Human Interaction*, pages 33–39, 1991.

[7] S.T. Dumais, G.W. Furnas, T.K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *CHI'88 Proceedings*, pages 281–285, 1988.

[8] M.E. Frisse and S.B. Cousins. Information retrieval from hypertext: Update on the dynamic medical handbook project. In *Proc. of the ACM Conference on Hypertext*, pages 199–212, 1989.

[9] D. Haines and W.B. Croft. Relevance feedback and inference networks. In *Proc. of the 16th Annual International ACM SIGIR Conference*, pages 2–10, 1993.

[10] C. Kaplan, J. Fenwick, and J.R. Chen. Adaptive hypertext navigation based on user goals and context. *User Modeling and User Adapted Interaction*, pages 193–220, 1993.

[11] P. Maes and R. Kozierok. Learning interface agents. In *Proc. of the 11th National Conference on Artificial Intelligence*, pages 91–99, 1993.

[12] N. Mathé and J.R. Chen. A user-centered approach to adaptive hypertext based on an information relevance model. In *Proc. of the Fourth International Conference on User Modeling*, pages 107–114, 1994.

[13] G. Salton. *Automatic Text Processing: the transformation, analysis, and retrieval of information by computers.* Addison Wesley, 1989.

[14] M. Schneider-Hufschmidt, T. Kuhme, and U. Malinowski, editors. *Adaptive User Interfaces - Principles and Practice.* North-Holland, 1993.

[15] J.W. Sullivan and S.W. Tyler, editors. *Intelligent User Interfaces.* ACM, 1991.